

# Flights Airlines Airports

Using Databricks, Spark, and Snowflake for  
flight data ETL and analysis



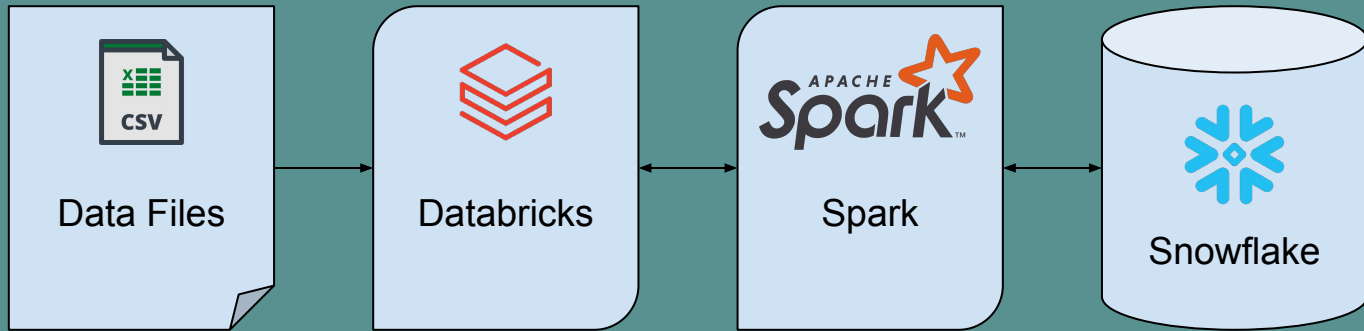
## Tools



**databricks**



# Data Flow



# Data

## CSV Files

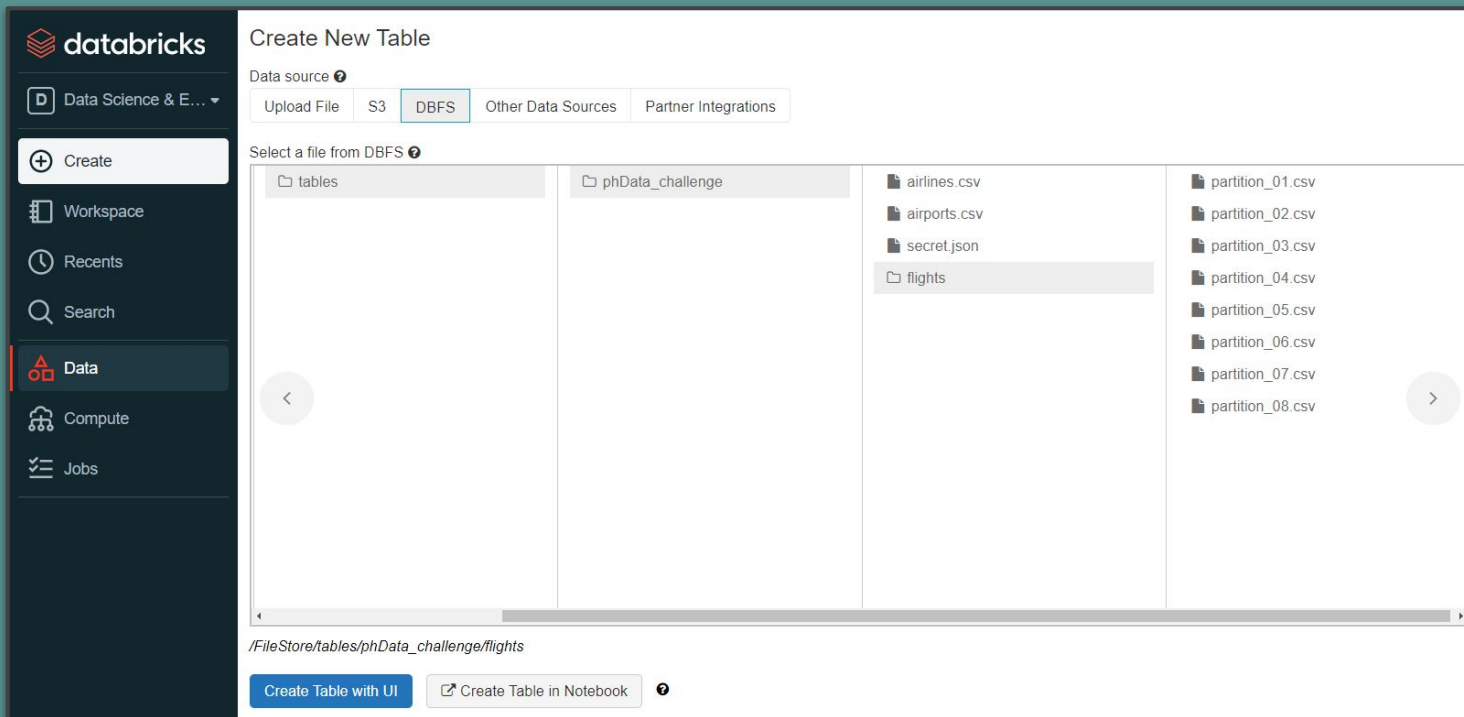
- 14 Airlines, 1 file
- 322 Airports, 1 file
- 3,920,766 Flights, 8 files



**3,920,766**  
**Flights!**

# Databricks

## DBFS contains CSV Files



The screenshot displays the Databricks web interface for creating a new table. On the left is a dark sidebar with the Databricks logo and navigation links: 'Data Science & E...', 'Create', 'Workspace', 'Recents', 'Search', 'Data' (highlighted), 'Compute', and 'Jobs'. The main panel is titled 'Create New Table'. Under 'Data source', tabs for 'Upload File', 'S3', 'DBFS' (selected), 'Other Data Sources', and 'Partner Integrations' are visible. Below this, a section 'Select a file from DBFS' shows a hierarchical file view. The path '/FileStore/tables/phData\_challenge/flight' is shown at the bottom. The file list includes 'airlines.csv', 'airports.csv', 'secret.json', 'flight' (selected), and a series of 'partition\_01.csv' through 'partition\_08.csv'. At the bottom, there are two buttons: 'Create Table with UI' and 'Create Table in Notebook'.

**databricks**

Data Science & E... ▾

+ Create

Workspace

Recents

Search

**Data**

Compute

Jobs

### Create New Table

Data source ⓘ

Upload File S3 **DBFS** Other Data Sources Partner Integrations

Select a file from DBFS ⓘ

tables phData\_challenge

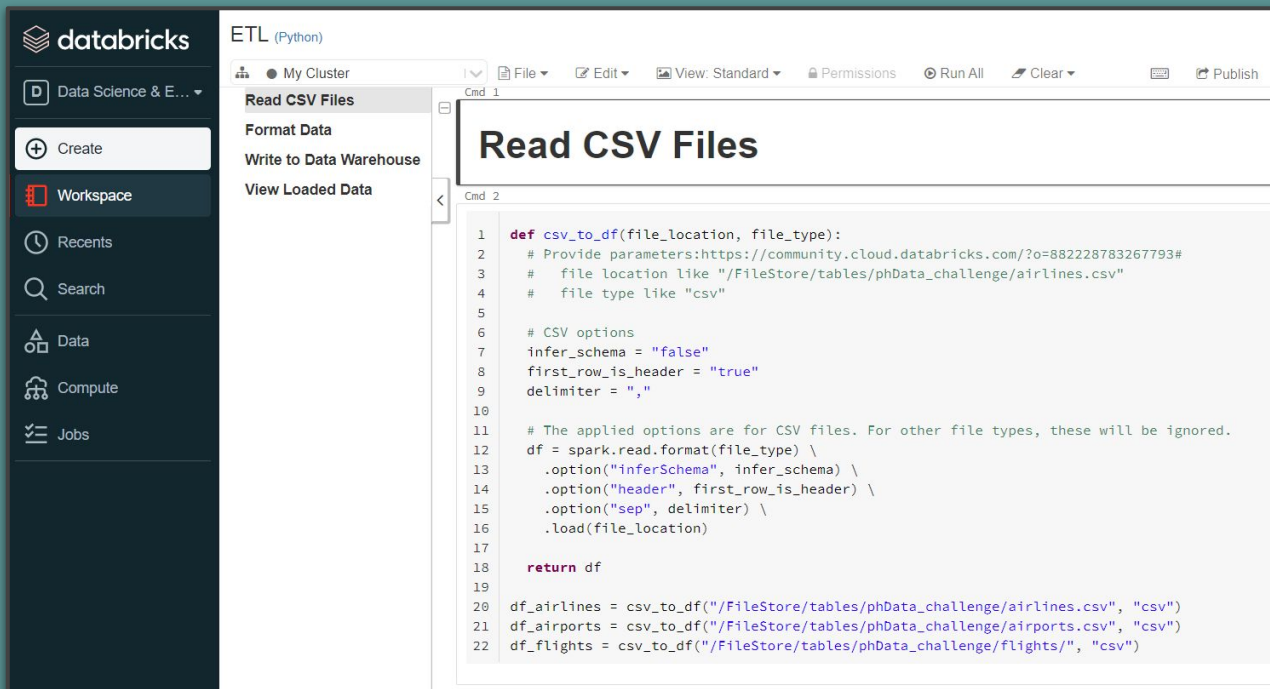
- airlines.csv
- airports.csv
- secret.json
- flight**
- partition\_01.csv
- partition\_02.csv
- partition\_03.csv
- partition\_04.csv
- partition\_05.csv
- partition\_06.csv
- partition\_07.csv
- partition\_08.csv

/FileStore/tables/phData\_challenge/flight

Create Table with UI Create Table in Notebook ⓘ

# Spark

## Extract, Transform, Load (ETL)



The screenshot displays the Databricks ETL (Python) interface. On the left is a dark sidebar with the Databricks logo and navigation options: Data Science & E..., Create, Workspace, Recents, Search, Data, Compute, and Jobs. The main workspace area is titled 'ETL (Python)' and shows a notebook named 'Read CSV Files'. The notebook's left sidebar lists actions: Read CSV Files, Format Data, Write to Data Warehouse, and View Loaded Data. The notebook content is divided into two command blocks. Command 1 is the title 'Read CSV Files'. Command 2 contains a Python function `def csv_to_df(file_location, file_type):` that reads CSV files with specific options and then uses `spark.read.format` to load the data into DataFrames for airlines, airports, and flights.

```
1 def csv_to_df(file_location, file_type):
2     # Provide parameters:https://community.cloud.databricks.com/?o=882228783267793#
3     # file location like "/FileStore/tables/phData_challenge/airlines.csv"
4     # file type like "csv"
5
6     # CSV options
7     infer_schema = "false"
8     first_row_is_header = "true"
9     delimiter = ","
10
11     # The applied options are for CSV files. For other file types, these will be ignored.
12     df = spark.read.format(file_type) \
13         .option("inferSchema", infer_schema) \
14         .option("header", first_row_is_header) \
15         .option("sep", delimiter) \
16         .load(file_location)
17
18     return df
19
20 df_airlines = csv_to_df("/FileStore/tables/phData_challenge/airlines.csv", "csv")
21 df_airports = csv_to_df("/FileStore/tables/phData_challenge/airports.csv", "csv")
22 df_flights = csv_to_df("/FileStore/tables/phData_challenge/flights", "csv")
```

# Snowflake

## Data Warehouse

The screenshot displays the Snowflake Data Warehouse web interface. The top navigation bar includes icons for Databases, Shares, Data Marketplace, Warehouses, Worksheets (active), History, Preview App, and Partner Connect. Below the navigation bar, the interface is divided into three main sections:

- Left Panel:** A sidebar for finding database objects. It shows a search bar with "Starting with..." and a tree view of the database structure. The tree is expanded to show the "Tables" section under the "PUBLIC" schema, listing tables like AIRLINES, AIRPORTS, CANCELLATION\_REASONS, and FLIGHTS. The "Views" section is also expanded, showing views like CANCELLATION\_REASONS\_BY\_AIR..., DELAYS\_BY\_AIRLINE, and FLIGHTS\_BY\_AIRLINE\_AIRPORT\_M...
- Center Panel:** A SQL editor window titled "delays\_by\_airline". It contains a SQL query that creates a view and selects data from it. The query is as follows:

```
1 -- Airline with the most unique routes (Top 1)
2 create or replace view most_unique_routes_airlines_top1 as
3 select top 1 *
4 from most_unique_routes_airlines
5 order by unique_routes desc;
6
7 select * from most_unique_routes_airlines_top1;
```
- Right Panel:** A results section titled "Results" with a "Data Preview" tab. It shows the execution status of the query, including a green checkmark, the query ID, the SQL statement, the execution time (1.6s), and the number of rows (1 row). Below this, there is a table with the following data:

Row	AIRLINE_CODE	AIRLINE_NAME	UNIQUE_ROUTES
1	EV	Atlantic Southeast Airlines	1351



# Visualization in Databricks

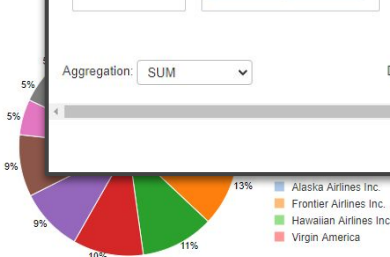
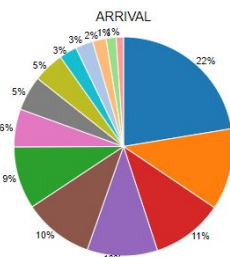
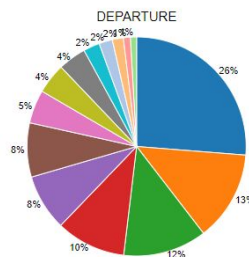
## Departure & Arrival Delays

Cmd 8

```
1 df = execute_query('''
2   select
3     airline_name as airlines,
4     departure_delays as departure,
5     arrival_delays as arrival,
6     total_delays as total
7   from delays_by_airline
8   order by total_delays desc
9   ''')
10 display(df)
```

▶ (1) Spark Jobs

▶ df: pyspark.sql.dataframe.DataFrame = [AIRLINES: string, DEPARTURE: decimal(18,0) ... 2 more fields]



## Customize Plot

All fields:

AIRLINES  
DEPARTURE  
ARRIVAL  
TOTAL  
<id>

Keys:

AIRLINES ✕

Series groupings:

<id>

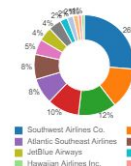
Values:

DEPARTURE ✕

ARRIVAL ✕

TOTAL ✕

DEPARTURE



ARRIVAL



TOTAL



Donut ☒

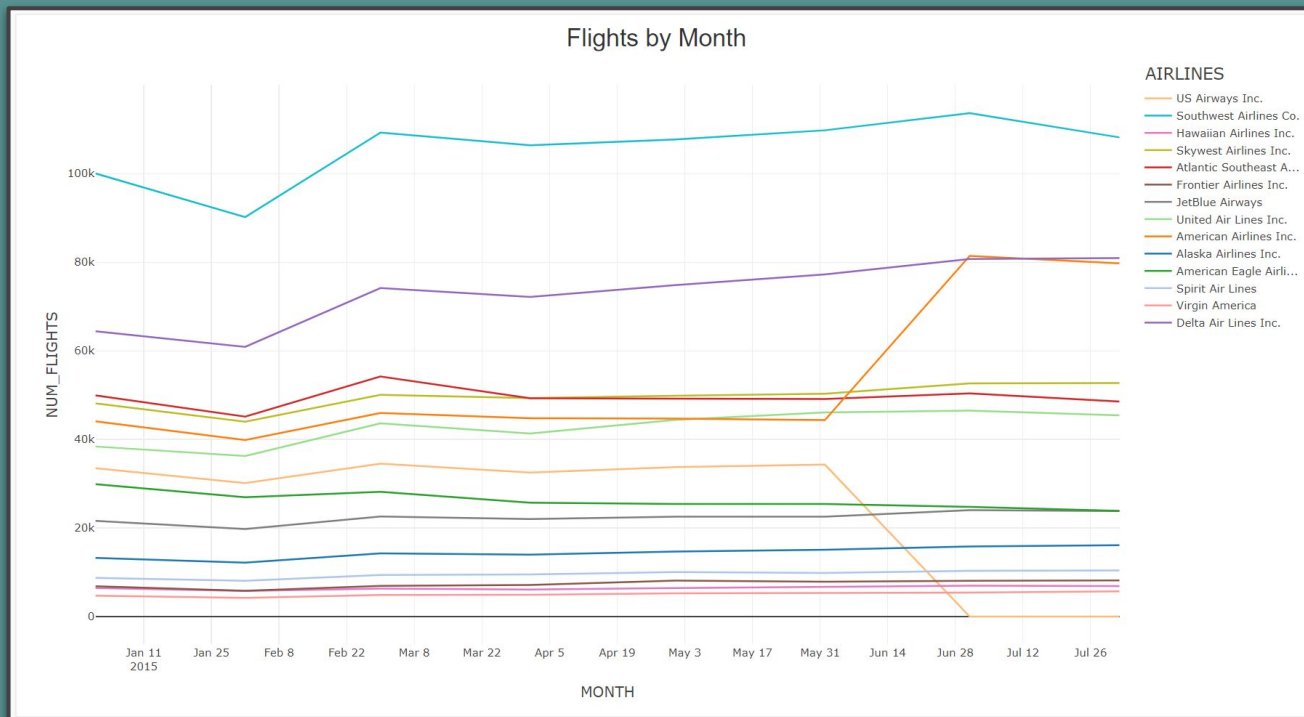
Aggregation: SUM

Display type: Pie chart

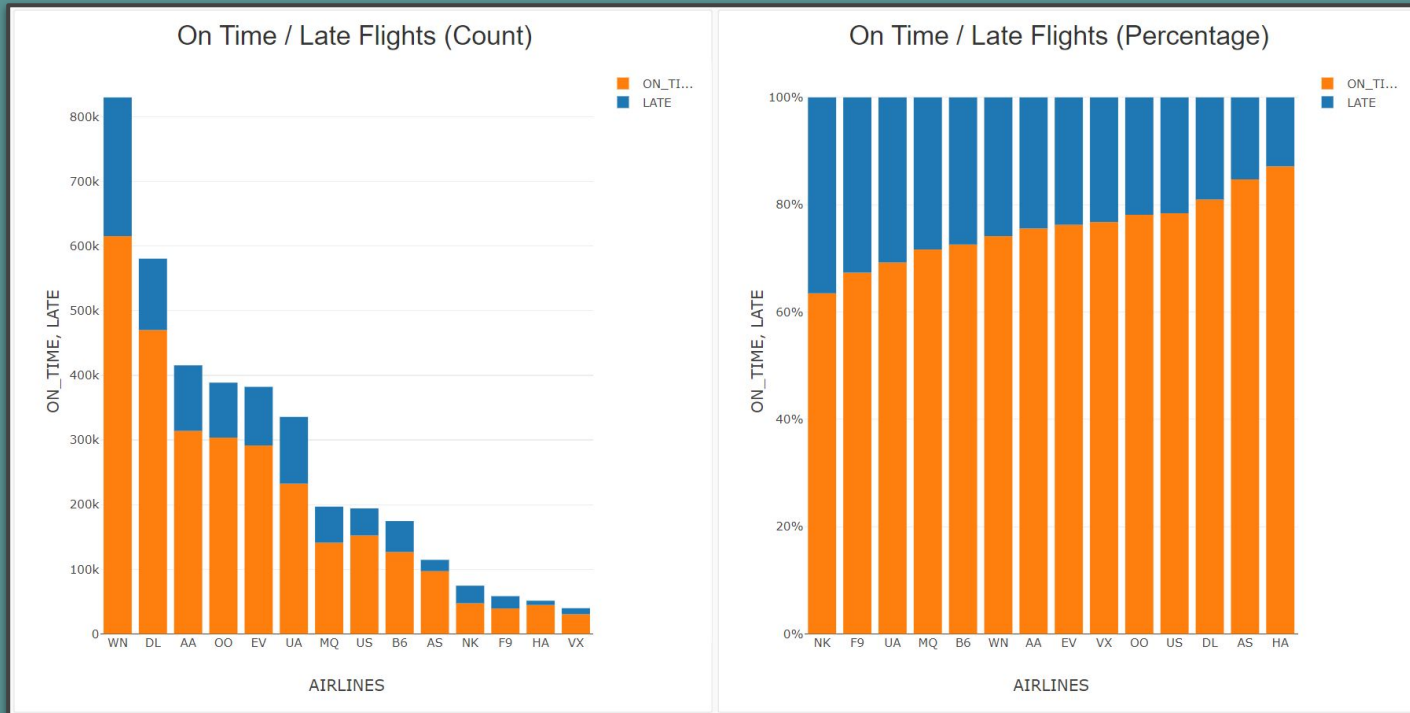
Cancel

Apply

Command took 1.74 seconds -- by bogdan.kovch@gmail.com at 6/10/2021, 4:51:03 PM on My Cluster

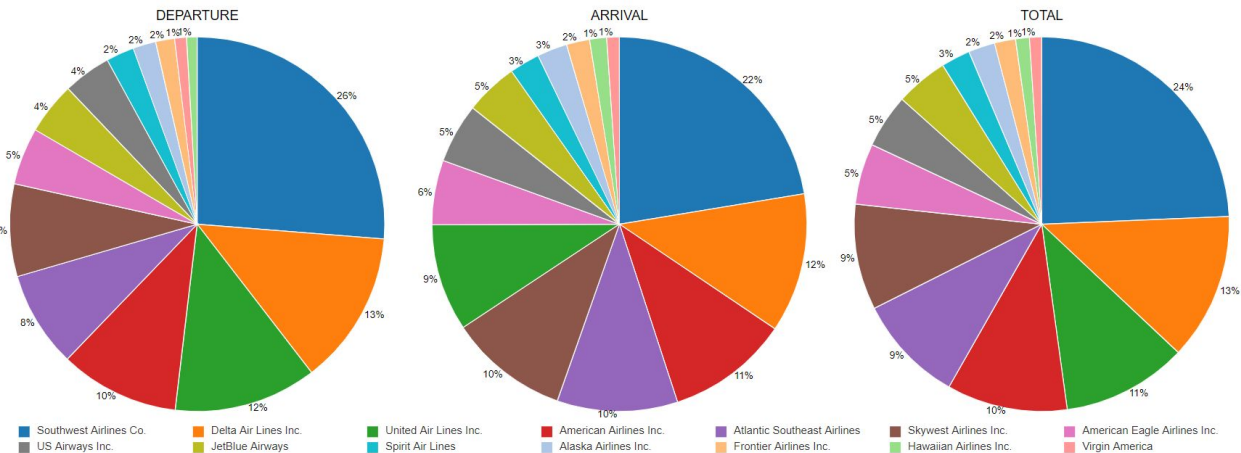


# On Time vs. Late Flights

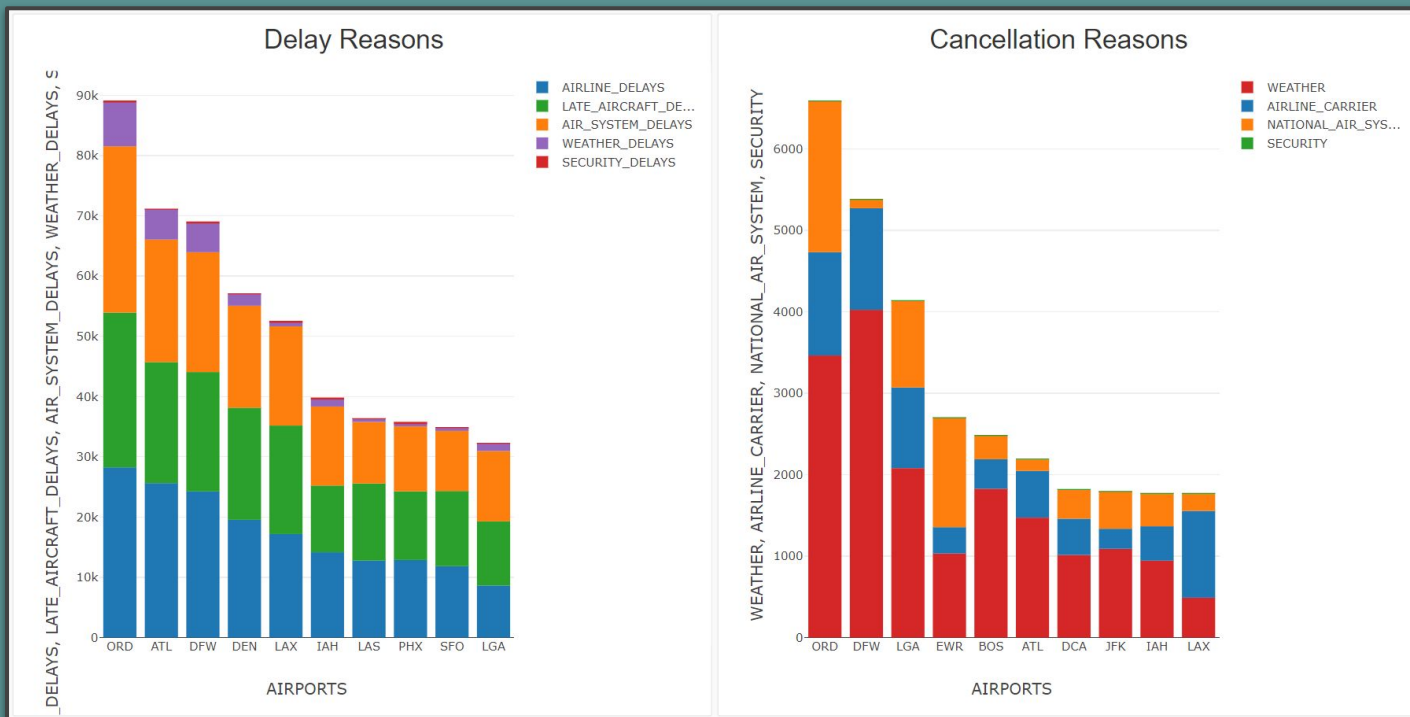


# Departure & Arrival Delays

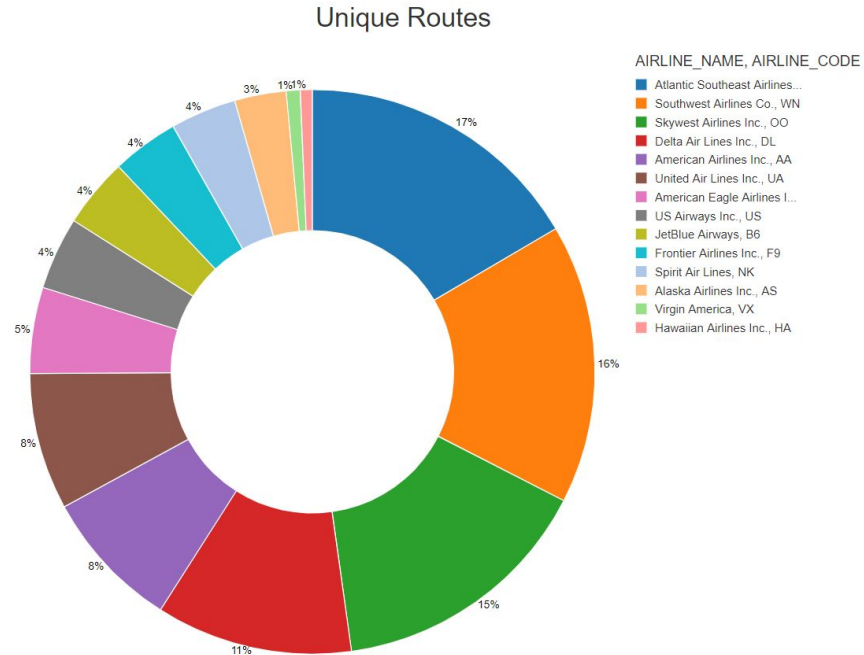
Departure & Arrival Delays



# Delay & Cancellation Reasons



# Variety of Routes





Prepared by Bogdan Kovch, June 2021

<https://github.com/bkovch/DataChallenge>

<https://www.linkedin.com/in/bogdan-kovch/>